Advanced Micro Devices

The Am9517A Multimode Direct Memory Access Controller

The Am9517A Multimode Direct Memory Access Controller

Advanced Micro Devices

The Am9517A Multimode Direct Memory Access Controller

AM-PUB092

TABLE OF CONTENTS

I. INTRODUCTION
- General 1
- Am9517A Overview 2

II. INTERFACING
- Block Diagram 3
- Interface Considerations 3

III. INTERFACE SIGNALS
- Clock 5
- Chip Select (CS) 5
- Reset (RST) 5
- Ready (RDY) 5
- Hold Acknowledge (HACK) 6
- DMA Request (DREQ0-DREQ3) 6
- Hold Request (HREQ) 6
- DMA Acknowledge (DACK) 6
- Address Bus (A0-A15) 6
- Address Strobe (ADSTB) 6
- Address Enable (AEN) 7
- Memory Read (MEMR) 7
- Memory Write (MEMW) 7
- IO Read (IOR) 7
- IO Write (IOW) 7
- End of Process (EOP) 7
- Data Bus (DB0-DB7) 7
- Power (VCC, VSS) 7

IV. REGISTER DESCRIPTION
- Address and Word Count Registers 8
- Status Register 9
- Command Register 9
- Request Register 10
- Mask Register 10
- Mode Register 11
- Byte Pointer Flip-Flop 12
- Temporary Data Register 12

V. SYSTEM INTERFACE
- Software Configurations 13
- General Control Routine 14
- Hardware Configurations 17

INTRODUCTION

General

Direct Memory Access (DMA), also sometimes known as "channel I/O" or "cycle stealing", has long been a feature of mini- and mainframe computer architectures. It was developed as a means to provide data transfer between system memory and peripheral devices at speeds higher than those obtainable under control of the CPU. This is achieved by bypassing the CPU and creating a direct path between memory and the peripheral device.

DMA capability significantly enhances the throughput performance of a processor system. To understand why this is so, recall that a CPU exercises its functions by reading an instruction from memory, decoding it, performing any address calculations necessary to locate operands, and then executing the specified operation. These are necessary steps with any processor, and it may take several instruction fetch/execute sequences to transfer each byte or word. For repetitive sequential data movement operations, the CPU can be removed from the transfer path and the normal fetch/execution steps simplified.

As an example, a data acquisition subroutine flow chart is illustrated in Figure 1. The subroutine stores bytes of data from an external source into successive memory locations. The main loop consists of several instructions requiring many clock cycles for each word transferred.

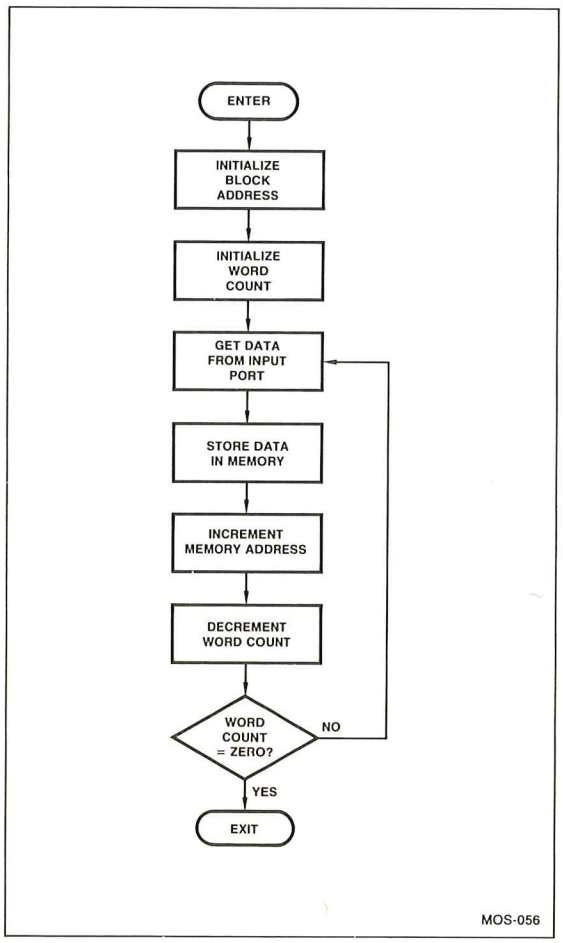


Figure 1. Block Transfer Flow Chart.

However, if what is required is to transfer a complete block of data between a source and a destination, a great deal of overhead can be eliminated by specialized hardware logic which provides access to system memory without CPU intervention. Direct Memory Access (DMA) consists of replacing the functions in the software loop with dedicated hardware. This can reduce the number of cycles required per word transferred and dramatically increase the information transfer rate.

For a more concrete comparison, the flow chart of Figure 1 has been coded in 8080A/8085 assembly language as shown in Figure 2. The transfer loop takes 10 bytes of program and executes in 46 clock cycles per byte. The same task executed in the Am9517A DMA Controller requires a little more initialization time but results in a transfer loop that takes only 3 clock cycles per byte (and an optional operating mode allows transfers only 2 clocks long for extra high speed). Thus, DMA operation can provide throughput gains of much more than an order of magnitude.

Architectural and semiconductor processing developments have occurred rapidly since the introduction of the first 8-bit general purpose microprocessor in 1973. Basic instruction execution times and minimum system component counts have both improved by more than an order of magnitude. CPU costs have fallen from several hundred dollars to well under ten dollars. The result has been an unprecedented growth in the applications for microprocessors. Furthermore, during this same interval, the 8-bit microprocessor that was originally conceived as a smart controller has evolved into a sophisticated computing element. This evolution has generated a concomitant need for efficient and high speed I/O transfers which is the forte' of direct memory access.

The ever increasing density of MOS/LSI has allowed the considerable amount of logic required within a useful direct memory access controller to be almost entirely incorporated on a single chip. Indeed, the only reason that more than one chip is required is the limitation on the number of pins available in industry-standard packages.

```
LOC   OBJ          SEQ      SOURCE STATEMENT
                    1 ;
                    2 ;     SUBROUTINE TO MOVE A BLOCK OF DATA
                    3 ;     FROM AN INPUT PORT INTO MEMORY
                    4 ;
                    5 ;
                    6 ;
                    7       EXTRN MEMA,CNT
                    8 ;
0001                9 PORT1 EQU 01H         ;INPUT PORT NO. 1
                   10 ;
                   11       CSEG
                   12 ;
0000  210000   E   13 BM17:  LXI H,MEMA     ;INIT. MEMORY ADDRESS
0003  010000   E   14        LXI B,CNT      ;INIT. WORD COUNT
0006  DB01         15 LOOP:  IN PORT1       ;READ FROM INPUT PORT
0008  77           16        MOV M,A        ;STORE DATA IN MEMORY
0009  23           17        INX H          ;INCREMENT MEMORY POINTER
000A  0B           18        DCX B          ;DECREMENT WORD COUNT
000B  78           19        MOV A,B        ;TEST IF WORD COUNT = 0?
000C  B1           20        ORA C
000D  C20600   C   21        JNZ LOOP       ;LOOP UNTIL DONE
0010  C9           22        RET
                   23 ;
                   24 ;
                   25        END
```

Figure 2. Programmed Block Transfer.

Various forms of DMA control have been implemented in the past. One approach is to interleave individual memory transfers with continuing CPU memory operations. This allows "transparent" DMA, but does not take advantage of the available memory bandwidth, plus the interface and control logic becomes very CPU-specific.

The Am9517A uses another technique where it takes over complete system control and can therefore make full use of the memory speed. This Application Note describes the Am9517A developed by Advanced Micro Devices. In addition to a full description of the part and its various modes of operations, implications and configuration suggestions are made to aid the system designer.

Am9517A Overview

The functional specification of the Am9517A Multimode Direct Memory Access Controller was developed to offer a very general DMA capability with a wide range of programmable options, many system-oriented features, and a general purpose interface facilitating the use of the device with a wide range of system architectures.

Careful study of the needs of a wide variety of applications which could benefit from DMA capability indicated the need for a minimum (but expandable) set of four separate and independent channels. Each channel in the Am9517A has associated with it two 16-bit registers which contain the current address and current word count information and two more registers which contain the base address and base word count. The base registers permit any channel to be automatically re-initialized at the end of a transfer. In addition, each channel has associated with it a 6-bit mode register which determines the types of transfers and options to be executed.

Overriding control of the operation of the DMA controller is provided by a master enable/disable bit in an internal command register. In addition, a four-bit mask register is provided to allow individual channels to be enabled and disabled. An important feature of the mask register is that its contents may be set or cleared in two ways. All four channels may be simultaneously enabled and/or disabled by means of a single command from the CPU. Alternatively, individual channels may be enabled or disabled without disturbing the status of the other channels. The use of this latter technique makes it unnecessary for the system software to maintain the mask status of all channels when changing just one.

Priority resolution logic is provided in order to resolve potential conflicts among requests for DMA service. Two software-selectable priority strategies are available to the system designer. Fixed priority gives the highest priority to channel 0 and the lowest to channel 3. Rotating priority maintains the same relative order as fixed priority, but assigns the lowest priority to the channel last serviced, thus preventing any one channel from monopolizing the controller.

Each of the channels operates in one of four modes. In the Single transfer mode one word is transferred in response to each DMA request of the channel. Block transfer mode causes the DMA controller to make continuous transfers until the word count for the active channel goes to 0. Demand transfer mode makes transfers as long as the request for DMA service is active and the word count for the channel is non-zero. The fourth mode of operation, called the cascade mode, is used to provide nearly unlimited expansion of the number of DMA channels available. A channel operating in cascade mode responds to a DMA request from a cascaded DMA controller by issuing a request to the CPU for control of the bus in the usual way. Acknowledgement by the CPU of the request is passed on to the cascaded controller by acknowledging the DMA request. No other address or control signals are activated by a cascade mode channel.

Two of the channels of the Am9517A can be used to provide memory-to-memory transfer capability. This feature offers very fast transfers of data blocks within system memory and is especially valuable in editing, initialization and other data movement operations. Channel 0 provides the source address while channel 1 provides the destination address and word count. A programmable feature of the memory-to-memory transfer operation is the ability to hold the source address constant. This allows the data contained in a single memory location to be replicated at very high speed throughout a block of memory.

Other programmable features which apply to all modes include the ability to select either incrementing or decrementing of the current address during transfers, a compressed timing feature which allows transfers to be executed in just two clock cycles, and the ability to select the active sense of the DMA Request and DMA Acknowledge signals.

A software DMA request capability is included which allows DMA transfers to be initiated by the processor on any channel. This powerful feature permits the processor itself to take full advantage of the capability of the DMA controller. An external hardware input is provided that allows the system to terminate a transfer when desired.

The Am9517A makes use of Advanced Micro Devices' LINOX N-channel silicon gate MOS technology. This process utilizes low profile structures, triple ion implantation, and both depletion and enhancement transistors to achieve very dense, high speed, low power circuitry. The chip contains 6350 transistors, has a total area of 41,580 square mils and is packaged in a standard 40-pin dual in-line package.

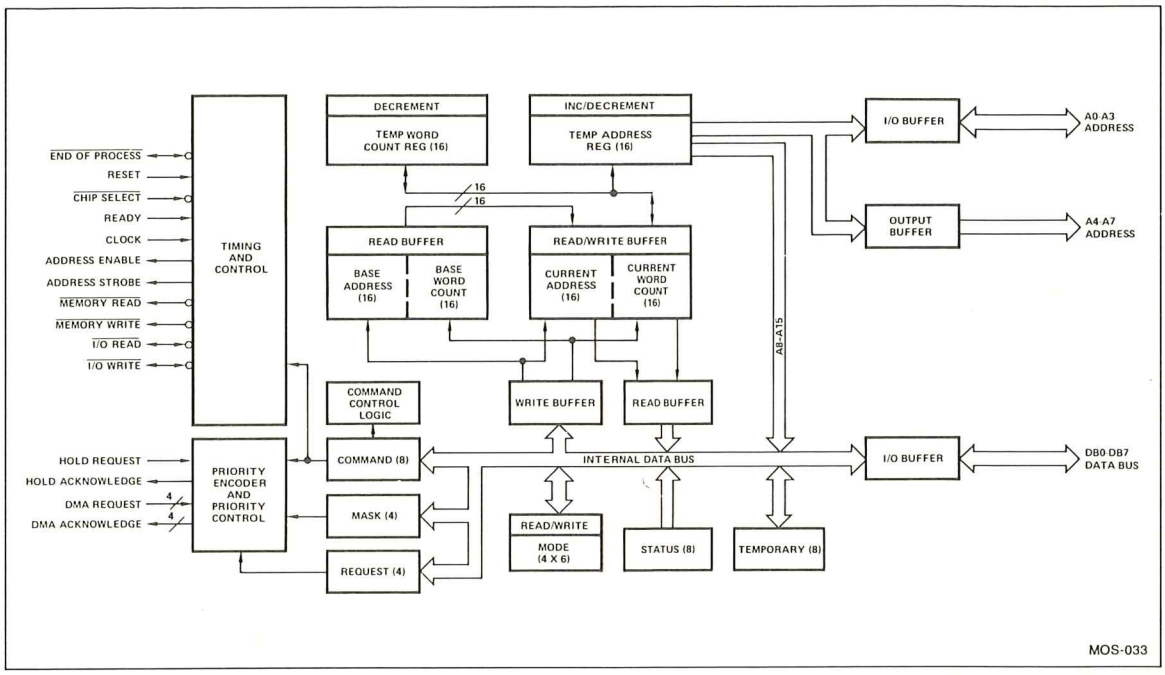


Figure 3. Am9517A Block Diagram.

INTERFACING

Block Diagram

The block diagram of the Am9517A (Figure 3) shows all of the interface signals in addition to the internal functional blocks and their data interconnections.

A peripheral device requiring service generates a DMA request to the Am9517A. If the channel receiving the request is enabled, a Hold Request to the system CPU is issued by the controller. When the CPU relinquishes control over the system busses, a Hold Acknowledge signal is output to the DMA controller to indicate that transfers may begin. On receipt of Hold Acknowledge the Am9517A issues a DMA Acknowledge to the highest priority, unmasked requesting device and begins issuing the control signals and addresses necessary to effect the desired transfers. Upon completion or termination of the transfer the Hold Request and DMA Acknowledge signals are terminated and the CPU regains control of the system busses. This procedure allows the DMA Controller to take full advantage of the available memory bandwidth and provides the greatest possible flexibility for transfer timing.

In order to establish the required operating characteristics within the DMA controller, internal registers are loaded under software control by the CPU. The numerous internal registers are addressed by means of the four least significant address lines (A0 through A3) which are thus made bidirectional. Address lines A4 through A7 are output by the controller and the eight high order address bits (A8 through A15) are demultiplexed from the data bus using the Address Strobe signal. The high order address byte is stored in an external latch and supplied to the address bus when required by the Address Enable (AEN) output signal. The I/O Read and I/O Write signals, in conjunction with Chip Select and the low order address bits, are used by the CPU to communicate with the controller.

During a DMA transfer the controller generates the combinations of read and write signals necessary to effect the transfers, using the Ready input, where necessary, to synchronize timing. End-Of-Process is a bidirectional signal which, as an output, indicates that a DMA transfer has been completed and, as an input, may be used to terminate any current transfer.

Interface Considerations

All of the input and output signals of the Am9517A are specified with worst-case levels identical to those of standard TTL circuits. Input logic levels are 2.0V high and 0.8V low; output logic levels are 2.4V high and 0.4V low. Thus, the normal worst-case noise immunity of 400mV offered by standard TTL logic is maintained. The logic level specifications take into account worst-case combinations of the three variables that effect the logic level thresholds: ambient temperature, supply voltage and processing parameters. Actual operating margins will be better than worst-case to the extent that these variables depart from worst-case conditions.

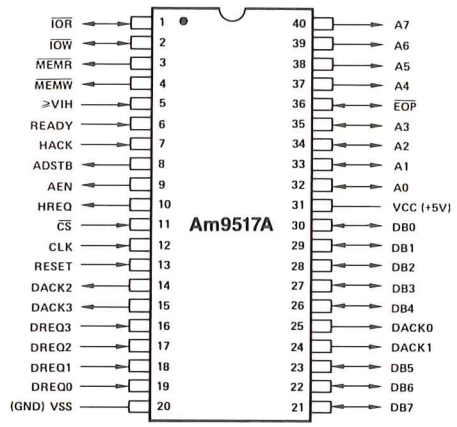
All outputs source at least 200 microamps at 2.4V. The Hold Request output is also specified for 100 microamps source current at 3.3V. All outputs sink a minimum of 3.2 milliamps at 0.4V. All the interface signals of the Am9517A are summarized by type in Figure 4a and their pin assignments are shown in Figure 4b.

The very high resistance of open gate MOS transistors exposes the device's input circuits to the risk of damaging accumulations of static charge. If charge enters the gate node of such an input faster than it can be discharged, the gate voltage can rise high enough to cause oxide breakdown, thus damaging or destroying the transistor.

All inputs to the Am9517A include protection networks designed to slow the transition times of incoming current surges and to provide low impedance discharge paths for voltages beyond normal

Signal Name	Abbreviation	Type	No. of Pins
Clock	CLK	Input	1
Chip Select	$\overline{CS}$	Input	1
Reset	RESET	Input	1
Ready	READY	Input	1
Hold Acknowledge	HACK	Input	1
DMA Request	DREQ0-DREQ3	Input	4
Hold Request	HREQ	Output	1
DMA Acknowledge	DACK0-DACK3	Output	4
Address Bus 4-7	A4-A7	Output	4
Address Strobe	ADSTB	Output	1
Address Enable	AEN	Output	1
Memory Read	$\overline{MEMR}$	Output	1
Memory Write	$\overline{MEMW}$	Output	1
I/O Read	$\overline{IOR}$	Input/Output	1
I/O Write	$\overline{IOW}$	Input/Output	1
Data Bus	DB0-DB7	Input/Output	8
Address Bus 0-3	A0-A3	Input/Output	4
End of Process	$\overline{EOP}$	Input/Output	1
+5 Volts	VCC	Power	1
Ground	VSS	Power	1

(a) Signal Summary.



(b) Connection Diagram.

MOS-034

Figure 4. Am9517A Interface Signals.

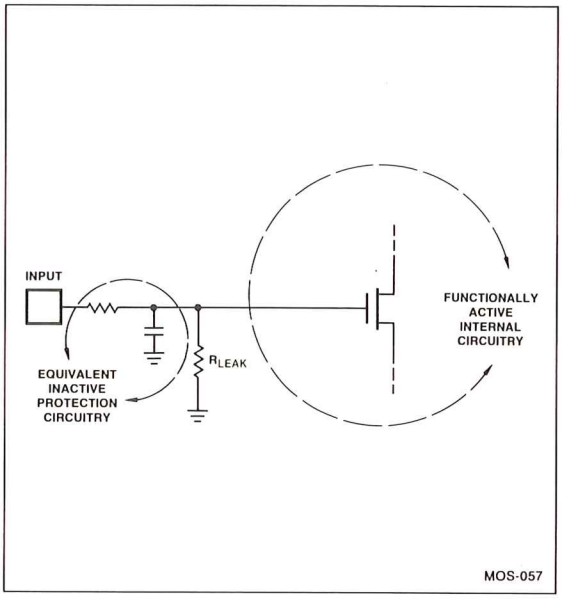


Figure 5. Input Circuitry.

operating levels. Note, however, that careless handling of MOS components can result in transfers of charge which cannot be absorbed without damage and conventional MOS handling precautions should be observed at all times.

In normal operation the input protection circuitry is inactive and may be considered as a lumped series RC network as shown in Figure 5. The active input connection during normal operation is the gate of an MOS transistor. No active sources or drains are connected to the inputs so that neither transient nor steady state currents are impressed upon the driving signals by the Am9517A The input signal is required only to charge or discharge the input capacitance and to overcome the leakage associated with the protection network and input circuit. Input capacitances are typically 6pF and leakage currents are usually less than 1μA. As is typical with MOS components, input drive specifications will usually be limited by transition time considerations rather than DC current limitations.

Good MOS design practice dictates that all inputs be terminated in order to provide discharge paths for transients. Unused inputs should be tied directly to ground or VCC as appropriate. Any input which is driven directly from a card edge connector should be terminated on the card in order to protect the input when the connection is broken. A simple pull-up resistor or on-board gate will suffice.

In general, on-chip delays will always track to a great extent and worst/best combinations will never occur together. The rising and falling edges of the read and write control pulses will track to provide minimum active widths.

INTERFACE SIGNALS

Clock

All the internal operations are clocked by this input which is used by the Am9517A to create four internal clocks. The number of operations to be performed within the controller and the high speed at which the device has been specified to operate, impose constraints on the input clock specification as illustrated in Figure 6. As long as the specified minimum high and low times are observed, the designer may use any convenient clock duty cycle. The greatest operating margins, and the best transition time tolerance, occurs when both clock low and clock high times are mutually maximized. Slowing the clock slightly, and thus extending TCY somewhat, also provides room for greater margins.

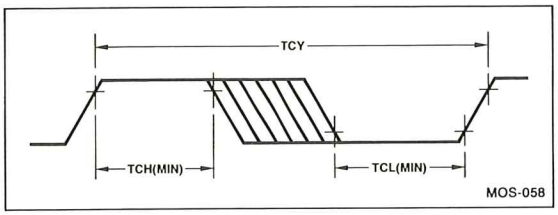


Figure 6. Clock Waveform Specifications.

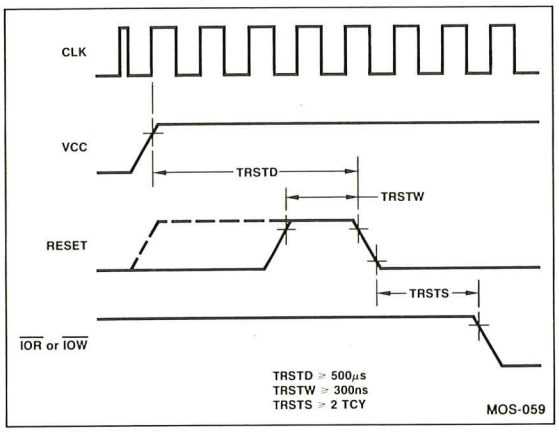


Figure 7. Reset Timing.

Chip Select ($\overline{CS}$)

Chip Select is an active-low input that enables data transfers between the DMA controller and the data bus. It is usually derived from an address output by the CPU to specify the device to be operated upon. The Chip Select input is recognized by the controller only when no active DMA or memory-to-memory transfers are taking place. It is ignored whenever the HACK input is true.

Reset

Reset is an active-high asynchronous input which terminates any operation that may be in progress within the controller. This input also clears the internal control registers, with the exception of the mask register which is set to all ones thus disabling all channels. After a reset the controller is ready to accept initialization commands from the CPU. Reset must be applied for a minimum of 300ns in order to insure that the controller is reset. The end of Reset should occur at least 500μs after both VCC and CLK are stable. At least two clock cycles must elapse following the end of Reset before the first I/O write is applied to the controller. Figure 7 shows the reset timing.

Ready

Ready is an active-high input which can be used to extend the read and write pulses generated by the DMA controller. If the Ready input is low throughout the Ready setup (TRS) and hold (TRH) times as shown in the left portion of Figure 8, the read and write pulses will be extended by one full clock cycle. The right portion of Figure 8 shows extension by two clocks. Ready is tested on the falling edge of each succeeding clock cycle and the read and write pulses are extended in multiple increments of TCY until Ready becomes true prior to the setup time. When utilizing the compressed timing feature (discussed later) to achieve a transfer rate of one word every two clock cycles, care must be taken to insure that Ready is true throughout the setup and hold time prior to the read and write pulses being true. Ready transitions should not occur during the defined setup-hold window in any operating mode.

In addition to extending the width of the Read and Write pulses, the Ready input going low causes the state of the address, data and control lines to be held constant. Note that, just as in the case of the CPU, the Ready input to the DMA controller remaining low for an extended period will hang up the system.

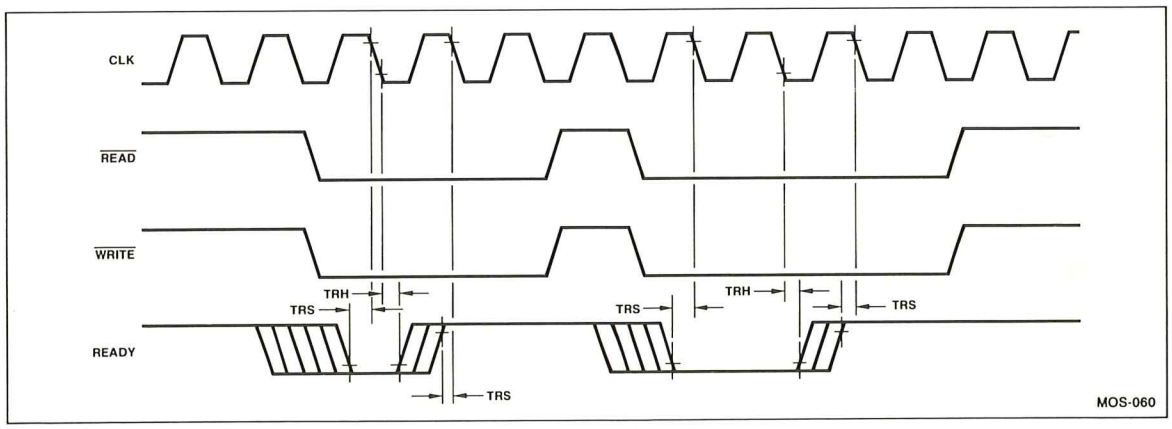


Figure 8. Ready Timing.

Hold Acknowledge (HACK)

The active-high Hold Acknowledge signals the controller that the system busses have been released by the CPU and placed in the high impedance state. This input initiates DMA transfers and should be asserted only in response to the issuance of a request for service by the Am9517A (HREQ). The DMA operations begin with the falling edge of the clock cycle following the satisfaction of the Hold Acknowledge setup time requirement (THS). The Hold Acknowledge input must remain true until the Am9517A relinquishes control of the system busses. The controller requires that one full clock cycle transpire between issuance of a HREQ and receipt of the Hold Acknowledge.

DMA Request (DREQ0-DREQ3)

The four asynchronous DMA Request inputs are used by peripheral devices to request DMA service. The polarity of the DREQ inputs is program selectable to be either active-high or active-low. A reset places them in the active-high condition.

An active DMA request is recognized by the Am9517A at the clock falling edge next following the satisfaction of the DREQ setup time (TQS). Any or all of the DMA request inputs may be active simultaneously and contention is resolved by priority logic contained within the DMA controller. Each Hold Acknowledge received by the controller selects the highest priority unmasked active DMA Request for the next transfer sequence. This means that a DMA transfer must be completed or otherwise terminated in order for a higher priority channel to be serviced.

Once asserted, an active DREQ input should be maintained at least until the corresponding DACK goes active. The implications of various methods of managing the DREQ signal timing after DACK is returned will depend on the operating mode being used for the associated channel.

Hold Request (HREQ)

The active-high Hold Request output indicates that the Am9517A requires service and is generated whenever an unmasked active DMA Request input is received. Timing of the Hold Request relative to the DREQ input is shown in Figure 9. In order to accommodate processors with higher level input specifications, the Hold Request output is designed to source at least 100 microamps at a VOH of 3.3 volts. The TDQ parameter shown in Figure 9 is specified at both 2.4 volts and 3.3 volts in the Am9517A data sheet. The higher level can be attained without assistance, but the timing specified requires an external pullup resistor.

DMA Acknowledge (DACK)

The four DMA Acknowledge outputs are each associated with one of the DMA Request inputs. Their active levels are selectable under program control. Unlike the DREQ inputs, after a reset the DMA Acknowledge outputs will be in the active-low condition. When a HACK is received from the CPU in response to the HREQ from the DMA controller, the highest priority, active, unmasked DMA Request will be granted service and the DMA Acknowledge output associated with that channel will become active, as shown in Figure 9. The DMA Acknowledge output from the Am9517A remains true until the completion of the requested DMA service and becomes inactive after the Hold Request output becomes inactive as shown in Figure 9. There will never be more than one DACK active at a time. No DACK is issued for memory-to-memory operations.

Address Bus (A0-A15)

In order to accommodate all of the functions provided by the Am9517A within the constraints of a 40-pin package, the 16 bits of address information provided by the device are output on two paths. The least significant eight are output on the eight address lines A0-A7. The four least significant address lines are bidirectional. As inputs they address the internal registers of the Am9517A when programming the device. Address lines A4 through A7 are tri-state outputs which are enabled only during DMA operations. The most significant eight bits of the address are output on the data bus during certain portions of the DMA operation. They can be demultiplexed from the data bus and stored in an external latch. The timing for the address bits and the two associated control lines is shown in Figure 10.

Address Strobe (ADSTB)

The active-high Address Strobe output from the Am9517A is the control signal used to latch the high order address bits (A8-A15) from the data bus into an external register or latch. Note that, as shown in Figure 10, the falling edge of Address Strobe should be used to clock the address bits into the latch; the data bus may not be valid at the rising edge. An important feature of the Am9517A is that the high order address and its associated strobe are issued only when required, namely during the first active cycle of a transfer and thereafter only when a carry or borrow is generated by the least significant address byte. This eliminates a clock cycle from the vast majority of transfers.

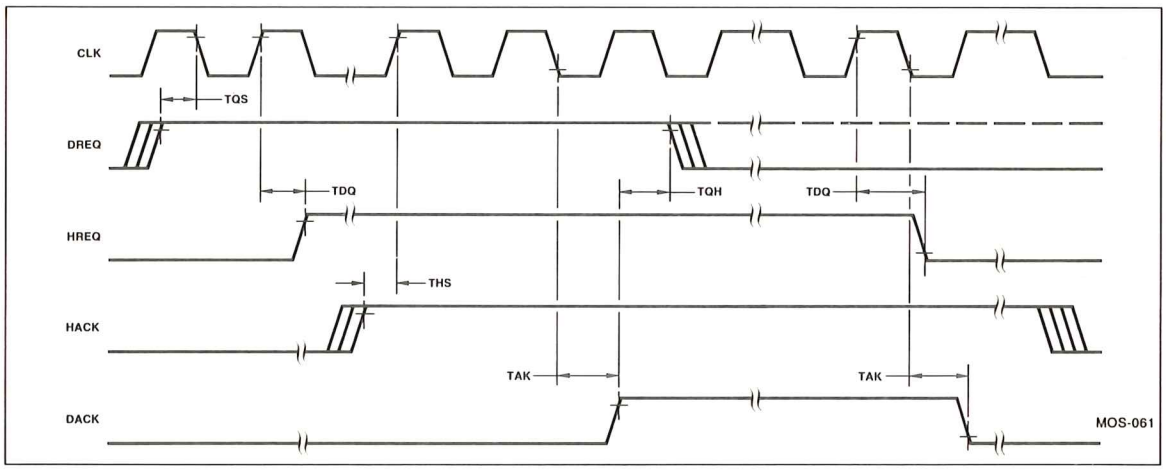


Figure 9. Peripheral and CPU Handshaking Interfaces.

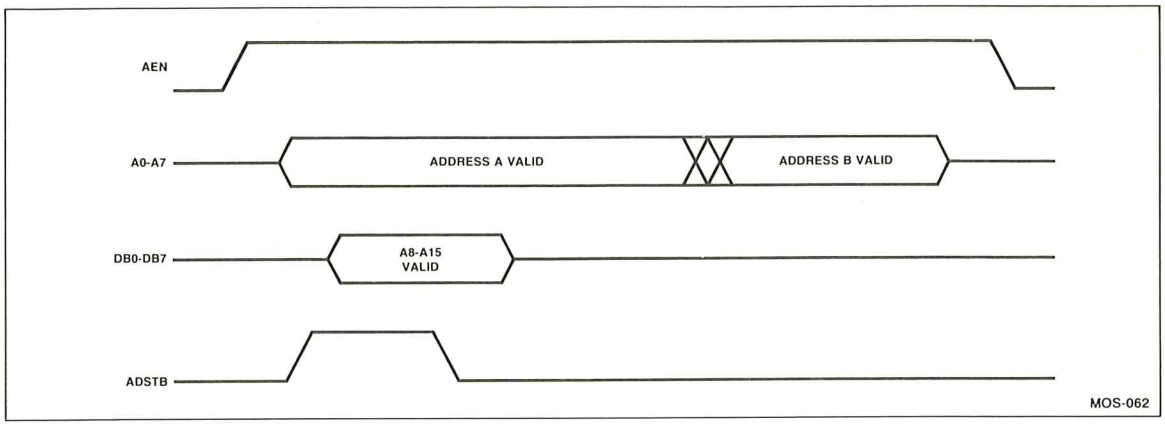


Figure 10. Address Control Relationships.

Address Enable (AEN)

Address Enable is an active-high control signal output by the Am9517A during the first clock cycle of a DMA operation. It remains valid throughout the transfer, as shown in Figure 10. The principal function of this signal is to enable the outputs of the tri-state latch which holds the eight high order address bits. Since the Address Enable output is asserted early in the DMA transfer and remains true until completion, it may also be used to put other system signals into their tri-state condition and to disable system activities such as IO chip select decoding. If AEN is true and all DACK remain false, a memory-to-memory operation is taking place.

Memory Read ($\overline{\text{MEMR}}$)

The transfer of data under the control of the Am9517A requires, in addition to the generation of address information, the provision of control signals to read data from a source location and write it into a destination. These control signals are derived from internally generated read and write signals based upon the type of transfer being executed. The active-low, three-state Memory Read output is used to initiate the reading of data from system memory.

Memory Write ($\overline{\text{MEMW}}$)

The active-low three-state Memory Write output is one of a group of four control signals utilized to initiate the reading and writing of data under the control of Am9517A. Its function is to control the writing of data into the system memory.

IO Read ($\overline{\text{IOR}}$)

IO Read is a bidirectional active-low signal. As an output it is used to control the reading of data from an external peripheral port. As an input, IO Read is recognized only if the DMA controller has been selected by the Chip Select input and no DMA operations are underway. If these conditions are met the IO Read input will cause an 8-bit byte to be read from the register addressed by A0-A3 onto the data bus.

IO Write ($\overline{\text{IOW}}$)

IO Write is a bidirectional active-low signal. As an output it is used to control the writing of data into peripheral ports. Like IO Read, this signal will not be recognized as an input unless CS is valid and no DMA operations are in progress. When recognized as an input this signal causes the information on the data bus to be loaded into the Am9517A register addressed by A0-A3.

$\overline{\text{IOR}}$, $\overline{\text{IOW}}$, $\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$ operate in pairs to control DMA information transfers. For peripheral-to-memory, $\overline{\text{IOR}}$ and $\overline{\text{MEMW}}$ are both active at the same time. For memory-to-peripheral, $\overline{\text{MEMR}}$ and $\overline{\text{IOW}}$ are both active in the same cycle. For memory-to-memory only $\overline{\text{MEMR}}$ and $\overline{\text{MEMW}}$ are used and only one at a time.

End of Process ($\overline{\text{EOP}}$)

End of Process is a bidirectional active-low signal. As an output it is active for a single clock period when the Word Count of an active DMA channel goes to zero. Asserting End of Process as an input causes the termination of DMA operations.

$\overline{\text{EOP}}$ may be used in many ways. As an interrupt, it can inform the CPU that a DMA transfer has occurred. It also can help coordinate peripheral device activity. When channel-specific signals are desired, $\overline{\text{EOP}}$ can be simply gated with individual DACK lines to generate EOP0-EOP3.

Data Bus (DB0-DB7)

The eight bidirectional three-state data bus signals transfer information between the Am9517A and the system data bus. During DMA operations the data bus signals are activated as outputs to supply the high order address byte. Note that the DMA data being transferred does not enter the Am9517A except in the case of memory-to-memory operation. During memory-to-memory operations the data being transferred is stored in a temporary register within the Am9517A between the read and write operations. During memory-to-memory operations the controller cycle which outputs the high order address byte and the address strobe is always present. During programmed IO operations on the Am9517A the data bus is utilized to transfer bytes between the CPU and the DMA controller.

Power (VCC, VSS)

The Am9517A makes use of a single +5 volt power supply and ground. One pin (pin 5) of the device which is not used by the interface must be at a logic high level. An internal pullup resistor is connected to pin 5 to provide the high level when the input is floated. Pin 5 may also be connected directly to VCC. The Am9517A requires a maximum of 150mA at 5V ±5% over the standard operating temperature range of 0°C to 70°C and 175mA at 5V ±10% over the full military temperature range of −55°C to +125°C. These maximum supply currents are worst-case values and apply at the lowest specified temperatures. Supply current always decreases with increasing temperatures; worst-case current at 25°C is 130mA.

REGISTER DESCRIPTION

Figure 11 summarizes the registers contained within the Am9517A. The various Address and Word Count registers control locations and numbers of transfers for active operations. The Command, Mode, Mask and Request registers manage the operating options and control features available to the system. Figure 12 shows the addressing used to access the Address and Word Count locations. Notice that A3 is always logic zero for that range of addresses. When A3 is a logic one, other locations are addressed as shown in Figure 13. All operations marked illegal should not be attempted as results will be unknown.

Name	Size	Number
Base Address Registers	16 bits	4
Base Word Count Registers	16 bits	4
Current Address Registers	16 bits	4
Current Word Count Registers	16 bits	4
Temporary Address Register	16 bits	1
Temporary Word Count Register	16 bits	1
Status Register	8 bits	1
Command Register	8 bits	1
Temporary Register	8 bits	1
Mode Registers	6 bits	4
Mask Register	4 bits	1
Request Register	4 bits	1

Figure 11. Am9517A Internal Registers.

Address and Word Count Registers

Each of the four channels provided by the Am9517A has associated with it four 16-bit programmable registers. Two of these, the current address and base address registers, are used to provide address information for data transfer. The current word count and base word count registers determine the number of words to be transferred by a DMA operation.

The base registers are loaded in parallel with the current registers. Two single-byte IO write operations to the same address are used to fill the 16-bit registers. An internal byte pointer flip-flop, which is cleared by a reset, master clear, or by IO command, changes state each time one of the 16-bit registers is accessed. It is used to steer the incoming 8-bit data to the least and most significant halves of the registers.

Two points must be emphasized with regard to the loading of the address and word count registers. First, since the current and base registers are loaded in parallel, no attempt should be made to change the contents of the base register while a DMA service is in progress on that channel. If, for example, transfer concatenation is desired, two channels should be utilized, with the incoming DMA requests being switched between them.

Second, the byte pointer flip-flop toggles automatically upon register access. This requires that care must be taken when accessing these registers other than in the initialization mode. For example, a subroutine which is called as a result of an interrupt and which accesses the address or word count registers should include the IO instruction which clears the byte pointer flip-flop.

Channel	Register	Operation	Interface Signals						Byte Pointer Flip/Flop	Data Bus DB0-DB7
			$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0		
0	Base & Current Address	Write	1 1	0 0	0 0	0 0	0 0	0 0	0 1	A0-A7 A8-A15
	Current Address	Read	0 0	1 1	0 0	0 0	0 0	0 0	0 1	A0-A7 A8-A15
	Base & Current Word Count	Write	1 1	0 0	0 0	0 0	0 0	1 1	0 1	W0-W7 W8-W15
	Current Word Count	Read	0 0	1 1	0 0	0 0	0 0	1 1	0 1	W0-W7 W8-W15
1	Base & Current Address	Write	1 1	0 0	0 0	0 0	1 1	0 0	0 1	A0-A7 A8-A15
	Current Address	Read	0 0	1 1	0 0	0 0	1 1	0 0	0 1	A0-A7 A8-A15
	Base & Current Word Count	Write	1 1	0 0	0 0	0 0	1 1	1 1	0 1	W0-W7 W8-W15
	Current Word Count	Read	0 0	1 1	0 0	0 0	1 1	1 1	0 1	W0-W7 W8-W15
2	Base & Current Address	Write	1 1	0 0	0 0	1 1	0 0	0 0	0 1	A0-A7 A8-A15
	Current Address	Read	0 0	1 1	0 0	1 1	0 0	0 0	0 1	A0-A7 A8-A15
	Base & Current Word Count	Write	1 1	0 0	0 0	1 1	0 0	1 1	0 1	W0-W7 W8-W15
	Current Word Count	Read	0 0	1 1	0 0	1 1	0 0	1 1	0 1	W0-W7 W8-W15
3	Base & Current Address	Write	1 1	0 0	0 0	1 1	1 1	0 0	0 1	A0-A7 A8-A15
	Current Address	Read	0 0	1 1	0 0	1 1	1 1	0 0	0 1	A0-A7 A8-A15
	Base & Current Word Count	Write	1 1	0 0	0 0	1 1	1 1	1 1	0 1	W0-W7 W8-W15
	Current Word Count	Read	0 0	1 1	0 0	1 1	1 1	1 1	0 1	W0-W7 W8-W15

Figure 12. Address for Word Count and Address Registers.

Interface Signals						Operation
A3	A2	A1	A0	$\overline{IOR}$	$\overline{IOW}$	
1	0	0	0	0	1	Read Status Register
1	0	0	0	1	0	Write Command Register
1	0	0	1	0	1	Illegal
1	0	0	1	1	0	Write Request Register
1	0	1	0	0	1	Illegal
1	0	1	0	1	0	Write Single Mask Register Bit
1	0	1	1	0	1	Illegal
1	0	1	1	1	0	Write Mode Register
1	1	0	0	0	1	Illegal
1	1	0	0	1	0	Clear Byte Pointer Flip/Flop
1	1	0	1	0	1	Read Temporary Register
1	1	0	1	1	0	Master Clear
1	1	1	0	0	1	Illegal
1	1	1	0	1	0	Illegal
1	1	1	1	0	1	Illegal
1	1	1	1	1	0	Write All Mask Register Bits

Figure 13. Register and Function Addressing.

Similarly, if only a single byte of these registers is to be accessed care must be taken to properly maintain the status of the byte pointer.

The Current Address register contains the memory address that is provided by the DMA controller during a transfer on that channel. The Current Address is automatically incremented or decremented, depending upon the programmable option selected, after each word is transferred and so, when read, will indicate the address of the next word to be transferred.

If the autoinitialization feature has been enabled for the channel, the end of a DMA operation, as indicated by either internal or external $\overline{EOP}$, will cause the contents of the Base Address register to be transferred to the Current Address register. Similarly, the Current Word Count register, which is decremented after each word is transferred, may be reinitialized with the contents of the Base Word Count register.

Status Register

This 8-bit read only register, which is accessed by an $\overline{IOR}$ at the address shown in Figure 13, provides the status of the four DREQ inputs and indicates whether a DMA operation has been completed. See Figure 14.

The four least significant bits of the status byte are individually set by an internal or external $\overline{EOP}$ signal. These bits are cleared by a reset, a master clear command, or by reading the status register. Since the status is not retained by the DMA controller after a read, system software will usually save all four of the least significant bits in order to insure that all completed DMA operations, and not merely the one of interest, may be detected.

The four high-order bits of the status register indicate the state of the four DREQ inputs. An active level at DREQ, independent of the programmed DREQ polarity, sets the status bit corresponding to the channel requesting service. Reading the status register has no effect upon these bits.

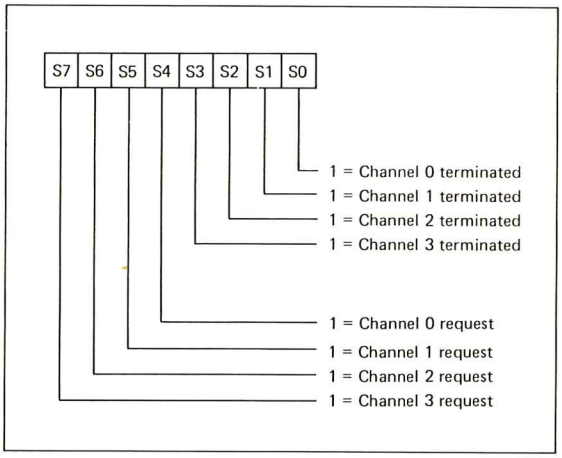


Figure 14. Status Register Bit Assignments.

Command Register

The 8-bit write-only command register is accessed at the location shown in Figure 13. It is cleared by a reset or master clear. Figure 15 indicates the functions carried out by each bit of the command register.

The least significant command bit (C0) controls the memory-to-memory feature. When performing memory-to-memory transfers the Channel 0 Address registers provide the source address, the Channel 1 Address registers provide the destination address and the Channel 1 Word Count registers determine the number of words to be transferred. Memory-to-memory operations are initiated by setting the software DREQ for channel 0.

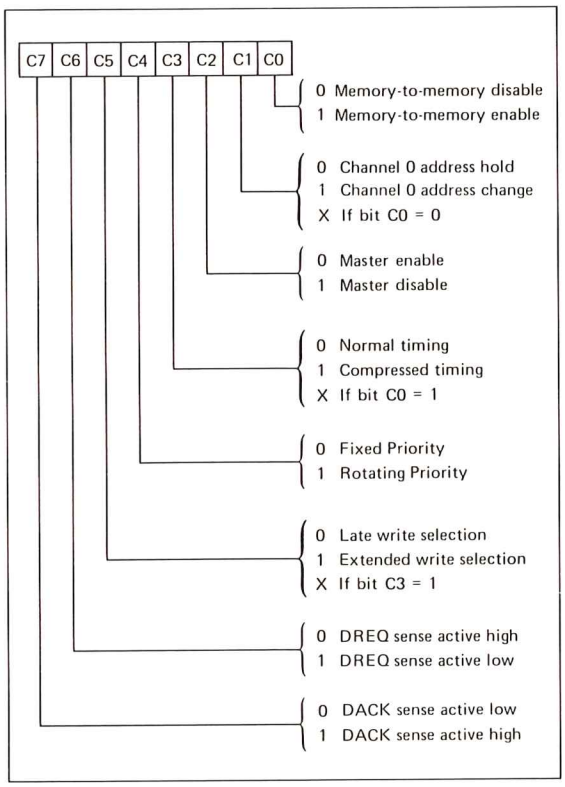


Figure 15. Command Register Bit Assignments.

Transfers proceed at a rate of 8 clock cycles per word. The transfers occur in block mode, that is to say, words will be transferred continuously until the Channel 1 word count reaches 0. It is recommended that Channels 0 and 1 be masked and that the Channel 0 Word Count be set to the same value as that in Channel 1 in preparation for memory-to-memory.

Command bit C1 is effective only if memory-to-memory transfers have been enabled. Under these conditions, setting this bit causes the normal incrementing or decrementing of the Channel 0 Current Address to be inhibited. The result is that the contents of the location defined by the Channel 0 Address register will be written throughout the block of memory defined by the Channel 1 Address and Word Count registers. This feature is useful in applications which require inserting spaces, building histograms and other multiple uses of a single character within data buffers for printers or displays. In some data acquisition applications it is useful to initialize a memory block with an offset value. In displays it will often be convenient to be able to clear screen very rapidly.

Command bit C2 controls the master enable/disable function. When this bit is set the HREQ output of the Am9517A is inhibited, thus preventing any DMA operations from occurring. Note that the entire command register is cleared by a reset or master clear operation, including C2. Although all the hardware DREQ inputs to the Am9517A are disabled by reset, software DMA requests are not masked and should be handled with care.

The compressed timing feature of the Am9517A is selected by means of Command bit C3. Normal timing uses three clock cycles per transfer except when the high-order address bits are output and four clocks are used. Compressed timing removes one clock cycle from each type of transfer thus shortening the transfers to two and three clocks. The effect of this is to reduce the read pulse width by TCY and to cause the read and write pulses to coincide. Where the requirements of system memory and the peripheral controller permit, a substantial increase in throughput can be obtained. This feature is not available during memory-to-memory transfers and the compressed timing bit is ignored if bit zero of the command word is set.

Bit C4 of the command word determines the type of priority arbitration to be utilized in resolving contending active DREQ inputs. If C4 is cleared, the four channels will be prioritized in fixed order, with channel 0 having the highest priority and channel 3 the lowest. If C4 is set, rotating priority is selected. The relative sequence of channel priorities will remain the same; however, upon completion of a DMA operation the most recently serviced channel will become the lowest priority. Thus if channel 2 where the last channel serviced, the order of priority would be channels 3, 0, 1 and 2. Rotating priority prevents a single channel from blocking service to other channels.

Priority arbitration is carried out as the first action of a DMA service upon receipt of the HACK input to the DMA controller. Thus, the highest priority unmasked DREQ input or software DMA request present at that time will be selected for service. Once a channel has been selected for service, control remains with that channel until the service is terminated. HREQ is relinquished following each service so reprioritization will occur.

When the cascade mode of operation is in use, service requests from cascaded DMA controllers are prioritized just like any other DREQ input before being passed on.

Bit C5 of the command register controls the Write pulse width and placement relative to the Read pulse. The optional extended write, which may be required for some memory systems or for some special IO requirements, is selected by this bit. The effect is similar to that of the compressed timing feature in that the write pulse becomes active simultaneously with the read pulse, although both are now two clock cycles wide. The extended write control bit is ignored with the compressed timing option is selected.

The last two bits (C6, C7) of the command register control the active levels of the DREQ inputs and DACK outputs. Note that the polarity of the two bits is such that DREQ will be active-high following reset and DACK will be active-low following reset.

Request Register

The 4-bit write-only Request register is accessed at the location shown in Figure 13. Each bit in the request register may be individually set or cleared based upon the state of the three least significant data bus bits. See Figure 16. The two least significant data bus bits select a channel while the third bit determines whether the selected bit is to be set or cleared. All four of the Request register bits are cleared by a reset or master clear. The request bit for an active channel is cleared by an $\overline{EOP}$.

Software DMA requests are nonmaskable but are disabled by the master disable bit in the command word and are subject to priority arbitration. Due to the nonmaskable nature of these requests, they should be issued only at the end of an initialization or other command sequence when all system setup is complete.

Mask Register

The four-bit write-only Mask register provides the capability to disable any or all of the hardware DREQ inputs to the Am9517A. In order to provide as much flexibility as possible in controlling the mask bits, two addresses are assigned, as shown in Figure 13.

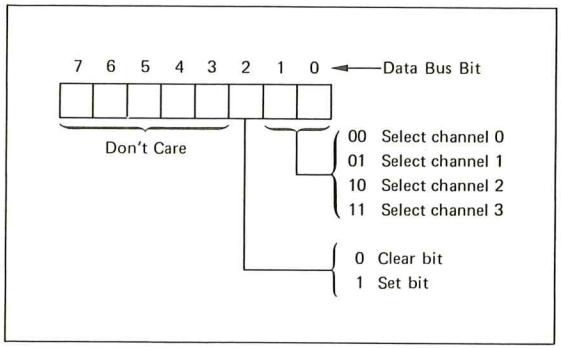


Figure 16. Single Bit Control for Request and Mask Registers.

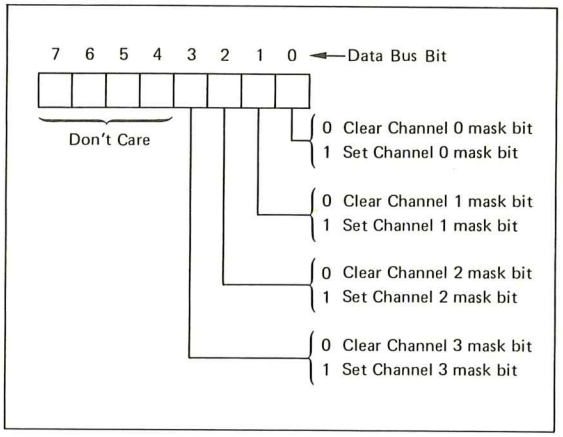


Figure 17. Parallel Mask Loading.

$\overline{IOW}$ commands enabled by $\overline{CS}$ and directed to address 1010 set or clear individual mask bits using the format of Figure 16. Alternatively, all four bits of the mask register may be written with a single command to address 1111. This command utilizes the four least significant data bus inputs to establish the status of the four DMA channels as shown in Figure 17.

In addition to program control of the mask register, the entire register is set by a reset or master clear thus disabling all external DMA requests. Furthermore, an $\overline{EOP}$ on an active channel not programmed to autoinitialize will set the corresponding bit in the mask register. This is done to prevent another DMA transfer from occurring before new address and word data have been set up. Software DMA requests are not maskable.

Mode Register

Each of the four DMA channels has an independent 6-bit write-only Mode register associated with it. This register is accessed at the location shown in Figure 13. When accessing the Mode registers the two least significant data bus inputs select the mode register to which the remaining 6 bits are to be transferred, as shown in Figure 18. These six bits determine the type of transfer, its mode, whether the transfers are to be in ascending or descending order, and whether autoinitialization is to be utilized.

Three types of transfer are selectable by the two least significant bits of the mode register (M2, M3). The verify transfer is a dummy operation during which all program selected functions of the Am9517A are executed except that the read and write control outputs are disabled. This permits the device and its addresses to be exercised without actually transferring data and can be used to verify proper operation of the controller.

The read transfer moves data from system memory to an IO device by activating the $\overline{MEMR}$ and $\overline{IOW}$ control outputs of the Am9517A. Write transfers utilize the $\overline{IOR}$ and $\overline{MEMW}$ control lines to move data from an IO device to system memory. The fourth possible combination of M2, M3 is an undefined state and should not be used. If the channel has been programmed for cascade operation the type of transfer is determined by the attached controller and bits M2, M3 are ignored.

Mode bit M4 specifies the autoinitialization option for the channel. When M4 = 0, the Current Address and Word Count registers are not affected by $\overline{EOP}$ and the associated mask bit is set, disabling the channel. When M4 = 1, occurrence of $\overline{EOP}$ causes the contents of the Base Address and Word Count registers to be transferred into the respective Current Address and Word Count registers. The mask bit is not set.

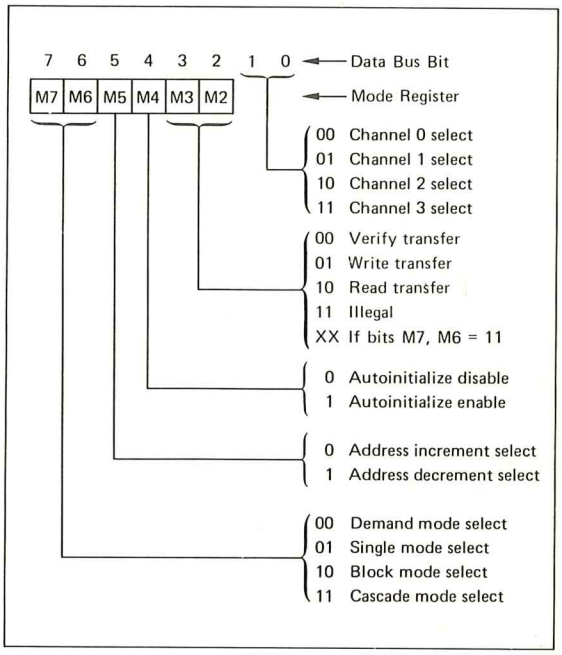


Figure 18. Mode Register Addressing and Assignments.

Autoinitialization allows repetitive DMA operations to proceed without software intervention between blocks. Since each channel has independent base registers, each can be independently autoinitialized without disturbing other channels.

Mode bit M5 specifies the address increment/decrement option for the channel. When M5 = 0 the address will increment following each transfer. When M5 = 1 the address will decrement following each transfer. This feature allows significant versatility in data movement. Blocks that may arrive from peripherals in reverse can be written in descending order in memory so that they end up being forward in memory. With the source incrementing and the destination decrementing, a memory-to-memory move can invert a list.

Memory-to-memory control can override the M5 bit for channel 0, forcing the address to neither increment nor decrement. See the Command Register description for details.

Mode bits M6 and M7 specify one of four available operating modes for the channel. These types of transfer management provide versatility for the interface between the peripheral device and the DREQ input to the Am9517A. One mode is also provided to greatly simplify expansion of the DMA system. All DMA transfers are initiated by an active-going DREQ signal. (Memory-to-memory transfers are initiated by a software request.) All DMA transfers are terminated by an internal (word count = 0) or external $\overline{EOP}$, and by a reset or master clear. The following mode descriptions indicate methods for managing transfers between initiation and final termination.

When M7,M6 = 00 Demand mode is selected. Once the DREQ has been accepted, continuous transfers will occur until the DREQ goes inactive (or until $\overline{EOP}$). When DREQ returns active, transfers will resume where they left off. This allows the requesting device to control the lengths of sub-block-sized bursts of transfers.

For example, if DREQ is cleared at the time that DACK is received, only one word will be transferred. Alternatively, if DREQ is cleared at the time that the internal $\overline{EOP}$ goes active, a complete block (as defined by the word count) will be transferred. Between those extremes Demand mode allows interaction of system activities with DMA transfers. A memory refresh cycle can be executed in the midst of a transfer; availability of external data can control the transfer duty cycle.

When M7,M6 = 01 the Single mode is selected. It operates in two ways. This mode always returns system control to the CPU following each word transferred. If DREQ is then inactive, transfers do not continue. If DREQ remains active transfers will continue (as long as word count is greater than zero) but will always be interleaved with a full HREQ/HACK handshake with the CPU. In the case of the 8080A, this means that machine cycles will alternate with transfers of single words. Notice that the recurring arrival of active HACK means that priority will be re-resolved after each Single transfer. If a higher priority request is pending, it will be serviced. Using rotating priority arbitration with several channels set up in Single mode would then interleave a transfer on each channel with CPU machine cycles.

When M7,M6 = 10, the Block mode is selected. Once the DREQ has been acknowledged (DACK), continuous transfers will take place until $\overline{EOP}$ occurs, independent of the state of the DREQ input. Thus a DREQ pulse wide enough to encompass DACK will cause the movement of an entire block of data. Memory-to-memory operations use Block mode exclusively.

When M7,M6 = 11, the channel may be used to cascade an additional Am9517A circuit in order to expand the channel capacity of the DMA system. Any channel or combination of channels at any level may be used for expansion. Two cascaded Am9517A chips provide a net of seven available channels.

A channel used for cascading simply provides access to the internal priority resolution circuitry by bypassing the other functions of the channel. Address, data and control outputs are disabled when a cascade channel is active, allowing the next active chip to be wire-ORed to the system signals.

Byte Pointer Flip/Flop

An internal flip/flop, toggled by each access to 16-bit registers, is used to select the most significant or least significant register byte. An $\overline{IOW}$ operation to address 1100 clears the pointer, causing it to select the least significant byte of the next 16-bit register accessed. The flip/flop is also cleared by a reset or by the master clear command.

Temporary Data Register

The 8-bit Temporary Data register is used during memory-to-memory transfers to provide temporary storage of the data being transferred. Each byte transferred remains in the temporary register until overwritten by the next and may be read out by the CPU. The Temporary Data register is not used during DMA transfers. It is cleared by a reset or a master clear.

Programming

The Am9517A will accept programming from the host processor any time that HACK is inactive; this is true even if HREQ is active. The responsibility of the host is to assure that programming and HACK are mutually exclusive. Note that a logical conflict can occur if the host masks out a DREQ that has just initiated a HREQ: when HACK arrives no valid request may be pending. To prevent this situation it is suggested that the controller be disabled before the channel mask is set, and then that the controller be re-enabled.

After power-up it is suggested that all internal locations, especially the Mode registers, be loaded with some valid value. This should be done even if some channels are unused.

SYSTEM INTERFACE

Software Configurations

Direct Memory Access is, by definition, transparent to the CPU and does not involve direct software. The programmer (and system designer) will, however, be concerned with initialization of the device, subsequent changes to its programmed configuration, and the potential effect on system activities of a device which is capable of suspending CPU operations for indeterminate times and is, when active, beyond control of the CPU.

This section presents a few routines as examples of ways to drive the Am9517A. They are shown in 8080A/8085 coding. The comments in the routines plus the surrounding text should provide sufficient guidance to allow translation into other machine languages without difficulty. Similarly, these approaches should be modified and extended to reflect the differing requirements of specific applications.

Figure 19 is an "EQU" table that defines the mnemonic labels used in the routines. The values are taken from Figures 12 and 13 and are written in binary to shown that correspondence. Some programmers may want to translate those entries to octal or hex values. Note that this table assumes that hardware IO port Chip Select decoding for the Am9517A selects zero for system address lines A4 through A7. For other arrangements the four high order bits in each EQU table entry should be changed to the value decoded.

Figure 20 shows a simple, straight-forward routine to configure the Am9517A as part of a system initialization procedure, perhaps following power-on for example. It shows the setup for a single channel – in this case channel 2 – but could easily be expanded to include more channels.

Interrupts are disabled to prevent other routines from possibly disturbing the byte pointer flip/flop. The command byte (line 50) specifies these options:

DACK active low
DREQ active low
Extended Write
Fixed Priority
Normal Timing
Controller Enabled
Address not held
Memory-to-memory off.

This routine does not use a software DMA request. If it did, the command byte might want to disable the controller until the initialization is complete in order to prevent unwanted transfers. Master Clear blocks hardware-originated DMA transfers by setting all mask bits, but does not prevent software-originated requests.

```
LOC     OBJ     SEQ             SOURCE STATEMENT
                  1 ;
                  2 ;
                  3 ;PORT ASSIGNMENTS FOR AM9517A A0-A3 INPUTS.
                  4 ;
                  5 ;ADDRESS AND WORD COUNT REGISTERS:
                  6 ;
0000              7 ADR0      EQU 00000000B     ;ADDRESS REGISTERS
0002              8 ADR1      EQU 00000010B
0004              9 ADR2      EQU 00000100B
0006             10 ADR3      EQU 00000110B
0001             11 WCT0      EQU 00000001B     ;WORD COUNT REGISTERS
0003             12 WCT1      EQU 00000011B
0005             13 WCT2      EQU 00000101B
0007             14 WCT3      EQU 00000111B
                 15 ;
                 16 ;CONTROL REGISTERS:
                 17 ;
0008             18 STAT      EQU 00001000B     ;STATUS REGISTER
0008             19 CMND      EQU 00001000B     ;COMMAND REGISTER
0009             20 RQST      EQU 00001001B     ;REQUEST REGISTER
000B             21 MODE      EQU 00001011B     ;MODE REGISTER
000D             22 TEMP      EQU 00001101B     ;TEMPORARY DATA REGISTER
000F             23 MSKR      EQU 00001111B     ;FULL MASK REGISTER
000A             24 MSKB      EQU 00001010B     ;SINGLE MASK BIT
                 25 ;
                 26 ;DIRECT COMMANDS:
                 27 ;
000C             28 CLBP      EQU 00001100B     ;CLEAR BYTE POINTER
000D             29 MCLR      EQU 00001101B     ;MASTER CLEAR
                 30 ;
                 31 ;
                 32 ;
```

Figure 19. EQU Table.

```
LOC     OBJ     SEQ         SOURCE STATEMENT
                42 ;
                43 ;
                44 ;SETUP ROUTINE FOR DMA CHANNEL #2
                45 ;
3000            46          ORG 3000H
                47 ;
3000 F3         48 STUP:    DI              ;DISABLE INTERRUPTS.
3001 D30D       49          OUT MCLR        ;MASTER CLEAR.
3003 3E62       50          MVI A,01100000B ;SELECT COMMAND OPTIONS,
3005 D308       51          OUT CMND        ;OUTPUT TO COMMAND PORT.
3007 3E9A       52          MVI A,10011010B ;SELECT MODE OPTIONS,
3009 D30B       53          OUT MODE        ;OUTPUT TO MODE PORT.
300B 3E00       54          MVI A,0         ;FORM LOWER BYTE,
300D D304       55          OUT ADR2        ;OUTPUT TO ADDRESS REG.
300F 3E0F       56          MVI A,0FH       ;FORM UPPER BYTE,
3011 D304       57          OUT ADR2        ;OUTPUT TO ADDRESS REG.
3013 3E07       58          MVI A,7FH       ;FORM LOWER BYTE,
3015 D305       59          OUT WCT2        ;OUTPUT TO WORD COUNT.
3017 3E00       60          MVI A,0         ;FORM UPPER BYTE,
3019 D305       61          OUT WCT2        ;OUTPUT TO WORD COUNT.
301B 3E02       62          MVI A,00000010B ;CLEAR CHANNEL #2
301D D30A       63          OUT MSKB        ;MASK BIT.
301F FB         64          EI              ;ENABLE INTERRUPTS.
3020 C9         65          RET             ;RETURN.
                66 ;
                67 ;
                68 ;
```

Figure 20. Simple SETUP Routine.

The mode byte (line 52) specifies these options:

Block transfer mode
Address increment
Autoinitialize
Read transfer.

With the autoinitialization option in effect, this channel will not have to be setup again until the channel function changes. The Read transfer moves data from the system memory out to the peripheral on Channel 2.

The memory addess of the first byte to be transferred is 0F00 hex. The number of bytes to be transferred is 007F hex. In both cases, the upper and lower bytes are output to the same port and are steered internally by the byte pointer flip-flop.

Once the channel is ready to go, its mask can be cleared (line 63) so that it will recognize a DREQ input. Interrupts are then enabled and control returned to the main program. If this routine, or a variation of it, is used as part of a more general system initialization program segment, it may be appropriate to not clear the mask (and possibly to not enable interrupts) until the complete system setup is finished. If this routine is embedded in the midst of application programming, it may be appropriate to omit the Master Clear function and instead to simply set the mask bit for the channel to be changed. In that case it would also be appropriate to clear the byte pointer before the address is output.

General Control Routine

The approach outlined in Figure 20 can be expanded, of course, to include as many channels as desired. In applications where there are variations in the setup data for a given channel, however, this scheme can be awkward. Figure 21 shows a more general design that provides more versatility for many applications.

The SDMA routine picks up the parameters it needs from the calling program. These "in-line" parameters are located immediately following the instruction that calls SDMA. The format is shown at the start of the routine. SDMA first pulls in the mode data which contains the channel number information as well. This is used to index into a branch table that then transfers control to the code segment appropriate for the channel. This portion of the routine then moves the Address and Word Count values from their in-line position into the DMA registers. SDMA assumes that the command configuration has already been established and will not be changed.

Notice that SDMA masks out the channel selected and then clears the mask after the changes are made. Some applications may want to wait until some other point in the program before clearing the mask.

Once SDMA is in place, other configurations and control sequences are easier to implement. For example, the STUP routine in Figure 20 can then be accomplished by inserting a CALL SDMA procedure at line 52 and eliminating lines 52 through 64.

```
LOC     OBJ     SEQ     SOURCE STATEMENT
                 75 ;
                 76 ;
                 77 ;GENERAL PURPOSE ROUTINE TO SETUP DMA CHANNELS
                 78 ;FOR THE AM9517A. SETUP PARAMETERS ARE PASSED
                 79 ;IN-LINE FROM THE CALLING PROGRAM.
                 80 ;
                 81 ;
                 82 ;
                 83 ;CALL FORMAT:
                 84 ;       CALL SDMA
                 85 ;       (MODE BYTE)
                 86 ;       (LOW ADDRESS BYTE)
                 87 ;       (HIGH ADDRESS BYTE)
                 88 ;       (LOW WORD COUNT BYTE)
                 89 ;       (HIGH WORD COUNT BYTE)
                 90 ;
                 91 ;
                 92 ;
3000             93         ORG 3000H
                 94 ;
                 95 ;
                 96 ;
3000 E3          97 SDMA:  XTHL          ;GET ADDRESS OF MODE BYTE.
3001 7E          98        MOV A,M       ;GET MODE BYTE.
3002 23          99        INX  H        ;POINT TO NEXT PARAMETER
3003 E3         100        XTHL          ;AND REPLACE ADDRESS.
3004 D30B       101        OUT  MODE     ;MODE BYTE TO MODE PORT.
3006 E603       102        ANI  03H      ;ISOLATE CHANNEL #
3008 F604       103        ORI  04H      ;FORM MASK BIT FORMAT AND
300A D30A       104        OUT  MSKB     ;MASK OUT THE BIT.
300C E603       105        ANI  03H      ;RE-ISOLATE CHANNEL # AND
300E 07         106        RLC           ;MULTIPLY BY 2.
300F 217C30     107        LXI  H,BTAB   ;GET BRANCH TABLE ADDRESS
3012 85         108        ADD  L        ;AND INDEX INTO TABLE BY
3013 6F         109        MOV L,A       ;TWICE THE CHANNEL #.
3014 D21830     110        JNC  BBB      ;COMPLETE 16 BIT ADDRESS
3017 24         111        INR  H        ;IF NECESSARY
3018 5E         112 BBB:   MOV E,M       ;USING INDEXED TABLE
3019 23         113        INX  H        ;POINTER, ASSEMBLE
301A 56         114        MOV D,M       ;BRANCH ADDRESS AND
301B EB         115        XCHG          ;MOVE IT INTO H,L.
301C D30C       116        OUT  CLBP     ;CLEAR BYTE POINTER.
301E F3         117        DI            ;DISABLE INTERRUPTS
301F E9         118        PCHL          ;BRANCH TO CODE SEGMENT
                119                      ;FOR SELECTED CHANNEL.
                120 ;
                121 ;
                127 ;
                128 ;
3020 E1         129 CH0:   POP  H        ;ADDRESS AND WORD COUNT
3021 7E         130        MOV A,M       ;PARAMETERS ARE
3022 D300       131        OUT  ADR0     ;FETCHED FROM CALLING
3024 23         132        INX  H        ;PROGRAM AND OUTPUT
3025 7E         133        MOV A,M       ;TO CHANNEL 0.
3026 D300       134        OUT  ADR0
3028 23         135        INX  H
3029 7E         136        MOV A,M
302A D301       137        OUT  WCT0
302C 23         138        INX  H
```

Figure 21. General Purpose SETUP Routine.

```
LOC   OBJ      SEQ        SOURCE STATEMENT
302D  7E       139        MOV A,M
302E  D301     140        OUT WCT0
3030  23       141        INX H
3031  FB       142        EI
3032  3E00     143        MVI A,0
3034  D30A     144        OUT MSKB        ;CLEAR #0 MASK.
3036  E9       145        PCHL            ;BRANCH TO CALLER.
               146 ;
               147 ;
3037  E1       148 CH1:   POP H           ;ADDRESS AND WORD COUNT
3038  7E       149        MOV A,M         ;PARAMETERS ARE
3039  D302     150        OUT ADR1        ;FETCHED FROM CALLING
303B  23       151        INX H           ;PROGRAM AND OUTPUT
303C  7E       152        MOV A,M         ;TO CHANNEL 1.
303D  D302     153        OUT ADR1
303F  23       154        INX H
3040  7E       155        MOV A,M
3041  D303     156        OUT WCT1
3043  23       157        INX H
3044  7E       158        MOV A,M
3045  D303     159        OUT WCT1
3047  23       160        INX H
3048  FB       161        EI
3049  3E01     162        MVI A,01H
304B  D30A     163        OUT MSKB        ;CLEAR #1 MASK.
304D  E9       164        PCHL            ;BRANCH TO CALLER.
               165 ;
               166 ;
               167 ;
               171 ;
               172 ;
               173 ;
304E  E1       174 CH2:   POP H           ;ADDRESS AND WORD COUNT
304E  7E       175        MOV A,M         ;PARAMETERS ARE
3050  D304     176        OUT ADR2        ;FETCHED FROM CALLING
3052  23       177        INX H           ;PROGRAM AND OUTPUT
3053  7E       178        MOV A,M         ;TO CHANNEL 2.
3054  D304     179        OUT ADR2
3056  23       180        INX H
3057  7E       181        MOV A,M
3058  D305     182        OUT WCT2
305A  23       183        INX H
305B  7E       184        MOV A,M
305C  D305     185        OUT WCT2
305E  23       186        INX H
305F  FB       187        EI
3060  3E02     188        MVI A,02H
3062  D30A     189        OUT MSKB        ;CLEAR #2 MASK.
3064  E9       190        PCHL            ;BRANCH TO CALLER.
               191 ;
               192 ;
               193 ;
3065  E1       194 CH3:   POP H           ;ADDRESS AND WORD COUNT
3066  7E       195        MOV A,M         ;PARAMETERS ARE
3067  D306     196        OUT ADR3        ;FETCHED FROM CALLING
3069  23       197        INX H           ;PROGRAM AND OUTPUT
306A  7E       198        MOV A,M         ;TO CHANNEL 3.
306B  D306     199        OUT ADR3
306D  23       200        INX H
```

Figure 21. General Purpose SETUP Routine. (Cont.)

```
LOC     OBJ     SEQ         SOURCE STATEMENT
306E 7E         201         MOV A,M
306F D307       202         OUT WCT3
3071 23         203         INX H
3072 7E         204         MOV A,M
3073 D307       205         OUT WCT3
3075 23         206         INX H
3076 FB         207         EI
3077 3E03       208         MVI A,03H
3079 D30A       209         OUT MSKB        ;CLEAR #3 MASK.
307B E9         210         PCHL            ;BRANCH TO CALLER.
                211 ;
                212 ;
                213 ;
307C 2030       214 BTAB:   DW  CH0         ;BRANCH TABLE
307E 3730       215         DW  CH1
3080 4E30       216         DW  CH2
3082 6530       217         DW  CH3
                218 ;
                219 ;
```

Figure 21. General Purpose SETUP Routine. (Cont.)

Hardware Configuration

Figure 22 shows the interconnections for interfacing a single Am9517A to a microprocessor system and an 'LS373. Other 8-bit latches may also be used and Figure 23 shows several other possibilities for both single and double controller configurations. The 'LS373 and 'LS374 parts offer small 20-pin packages and low power. Figure 24 is a block diagram of the general Am9517A expansion scheme.

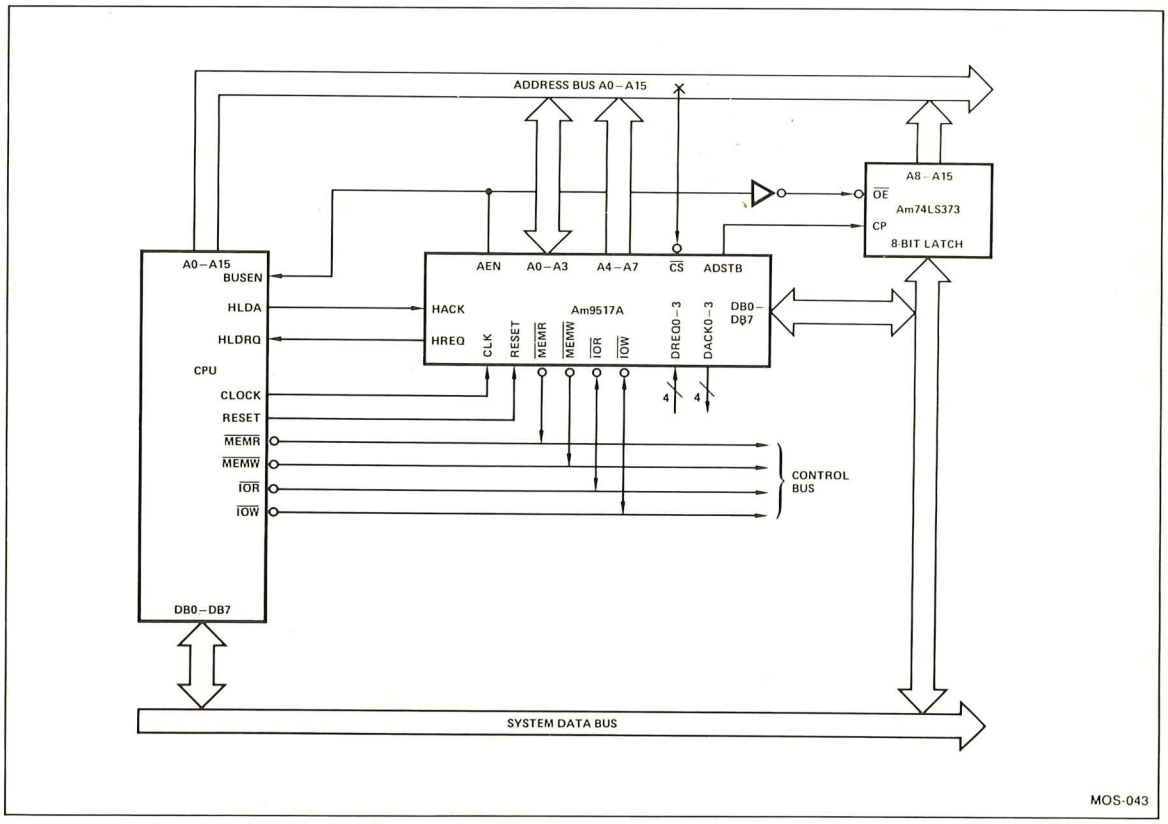


Figure 22. Basic DMA Configuration.

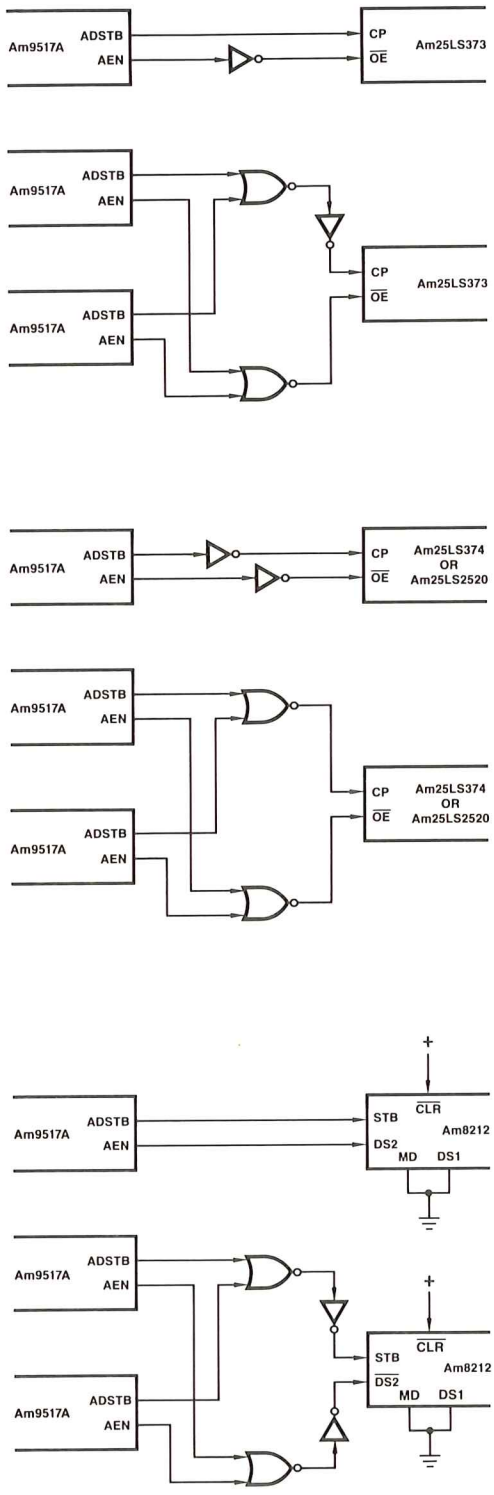

Am9517A
ADSTB
AEN
CP
OE
Am25LS373
Am9517A
ADSTB
AEN
Am9517A
ADSTB
AEN
CP
OE
Am25LS373
Am9517A
ADSTB
AEN
CP
OE
Am25LS374
OR
Am25LS2520
Am9517A
ADSTB
AEN
Am9517A
ADSTB
AEN
CP
OE
Am25LS374
OR
Am25LS2520
+
Am9517A
ADSTB
AEN
STB
CLR
DS2
Am8212
MD
DS1
Am9517A
ADSTB
AEN
+
Am9517A
ADSTB
AEN
STB
CLR
DS2
Am8212
MD
DS1
MOS-064

Figure 23.

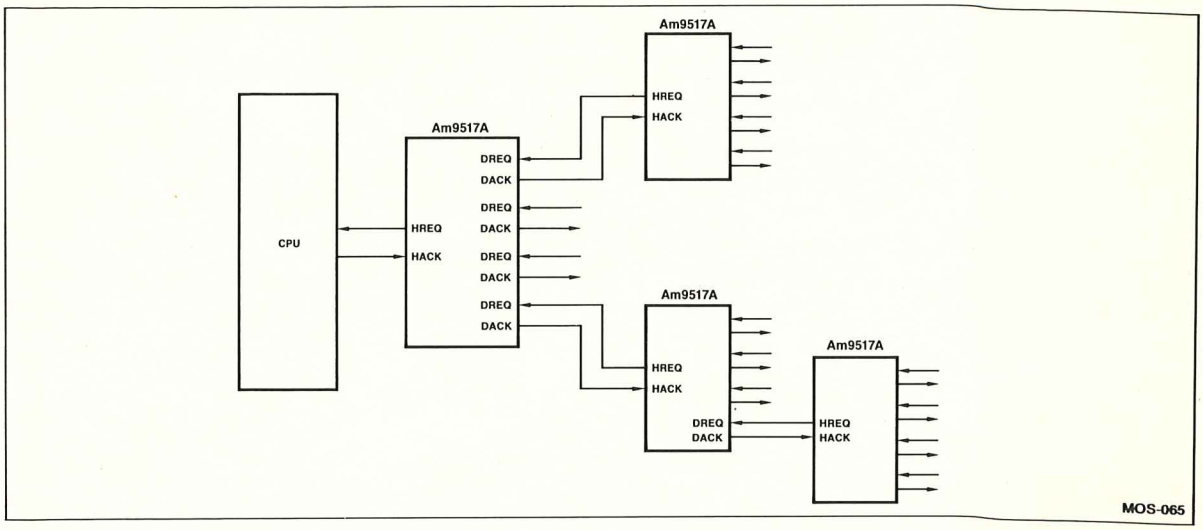
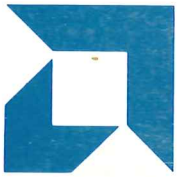

Am9517A
HREQ
HACK
Am9517A
DREQ
DACK
DREQ
DACK
DREQ
DACK
DREQ
DACK
CPU
HREQ
HACK
Am9517A
HREQ
HACK
DREQ
DACK
Am9517A
HREQ
HACK
MOS-065

Figure 24.



ADVANCED MICRO DEVICES, INC.
901 Thompson Place
Sunnyvale
California 94086
(408) 732-2400
TWX: 910-339-9280
TELEX: 34-6306
TOLL FREE
(800) 538-8450

10-79